

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In Re Application of:

Martin, et al.

Serial No.: 10/630,071

Filed: July 30, 2003

Group Art Unit: 2144

Examiner: Ibrahim, Mohamed

Docket No. 200208611-1

For: **Systems and Methods for Collecting Data Regarding a Messaging Session**

**APPEAL BRIEF UNDER 37 C.F.R. § 41.37**

Mail Stop: Appeal Brief-Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, Virginia 22313-1450

Sir:

This Appeal Brief under 37 C.F.R. § 41.37 is submitted in support of the Notice of Appeal filed October 30, 2008, responding to the final Office Action mailed September 16, 2008.

It is not believed that extensions of time or fees are required to consider this Appeal Brief. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required therefor are hereby authorized to be charged to Deposit Account No. 08-2025.

### **I. Real Party in Interest**

The real party in interest is Hewlett-Packard Development Company, LP, a limited partnership established under the laws of the State of Texas and having a principal place of business at 20555 S.H. 249 Houston, TX 77070, U.S.A. (hereinafter "HPDC"). HPDC is a Texas limited partnership and is a wholly-owned affiliate of Hewlett-Packard Company, a Delaware Corporation, headquartered in Palo Alto, CA. The general or managing partner of HPDC is HPQ Holdings, LLC.

### **II. Related Appeals and Interferences**

There are no known related appeals or interferences that will affect or be affected by a decision in this Appeal.

### **III. Status of Claims**

Claims 5-7, 10, 12-20, 22, and 26-36 have been canceled leaving claims 1-4, 8, 9, 11, 21, and 23-25 remaining. Each of those claims stand finally rejected. No claims have been allowed. The final rejections of claims 1-4, 8, 9, 11, 21, and 23-25 are appealed.

### **IV. Status of Amendments**

No amendments were made subsequent to the final Office Action. The claims in the attached Claims Appendix (see below) reflect the present state of the claims.

## **V. Summary of Claimed Subject Matter**

The claimed inventions are summarized below with reference numerals and references to the written description ("specification") and drawings. The subject matter described in the following appears in the original disclosure at least where indicated, and may further appear in other places within the original disclosure.

Independent claim 1 describes a method for collecting data regarding a messaging session. The method comprises intercepting an incoming message sent to a first network service. *Applicant's specification*, page 21, lines 11-14; Figure 7, item 700. The method of claim 1 further comprises writing a session identifier to a thread-local variable, the session identifier identifying a messaging session to which the incoming message relates. *Applicant's specification*, page 21, lines 22-23; Figure 7, item 704. The method of claim 1 further comprises storing in a database in relation to the session identifier session data relevant to the incoming message, the session data at least including a message received time. *Applicant's specification*, page 14, lines 6-9; page 22, lines 14-16; Figure 7, item 706. The method of claim 1 further comprises providing the incoming message to the first network service. *Applicant's specification*, page 22, lines 17-18; Figure 7, item 708. The method of claim 1 further comprises sending an outgoing message from the network service to a second network service or a client. *Applicant's specification*, page 22, lines 19-21. The method of claim 1 further comprises intercepting the outgoing message sent by the first network service. *Applicant's specification*, page 22, lines 21-22; Figure 8, item 800. The method of claim 1 further comprises performing a thread-local variable lookup so as to retrieve the session identifier written to the thread-local variable. *Applicant's specification*, page 22,

line 22 to page 23, line 1; Figure 8, item 802. The method of claim 1 further comprises storing in the database in relation to the session identifier session data relevant to the outgoing message, the session data at least including a message sent time. *Applicant's specification*, page 13, lines 8-12; page 23, lines 6-8; Figure 8, item 806. The method of claim 1 further comprises providing the outgoing message to the second network service or client. *Applicant's specification*, page 23, lines 8-10; Figure 8, item 808.

Independent claim 21 describes a physical computer-readable medium that stores a system for collecting data regarding a messaging session. The medium of claim 21 comprises logic configured to intercept an incoming message directed at a network service. *Applicant's specification*, page 21, lines 11-14; Figure 7, item 700. The medium of claim 21 further comprises logic configured to identify a session identifier that identifies a messaging session to which the incoming message relates. *Applicant's specification*, page 21, lines 14-17; Figure 7, item 702. The medium of claim 21 further comprises logic configured to write the session identifier to a thread-local variable. *Applicant's specification*, page 22, line 22 to page 23, line 1; Figure 8, item 802. The medium of claim 21 further comprises logic configured to store in a database in relation to the session identifier session data relevant to the incoming message, the session data at least including a message received time. *Applicant's specification*, page 14, lines 6-9; page 22, lines 14-16; Figure 7, item 706. The medium of claim 21 further comprises logic configured to intercept an outgoing message sent by the network service. *Applicant's specification*, page 22, lines 19-21. The medium of claim 21 further comprises logic configured to perform a thread-local variable lookup so as to retrieve the session identifier written to the thread-local variable. *Applicant's specification*, page

22, line 22 to page 23, line 1; Figure 8, item 802. The medium of claim 21 further comprises logic configured to store in the database in relation to the session identifier session data relevant to the outgoing message, the session data at least including a message received time. *Applicant's specification*, page 14, lines 6-9; page 23, lines 8-10; Figure 8, item 808.

## **VI. Grounds of Rejection to be Reviewed on Appeal**

The following grounds of rejection are to be reviewed on appeal:

1. Claims 1-4, 8, 9, 11, 21, and 23-25 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over *Karakashian, et al.* ("Karakashian," U.S. Pub. No. 2004/0064503) in view of *Kaler, et al.* ("Kaler," U.S. Pub. No. 2004/0199586) and further in view of *Kennedy* (U.S. Pat. No. 6,330,589).

## **VII. Arguments**

The Appellant respectfully submits that Applicant's claims are not obvious under 35 U.S.C. § 103, and respectfully requests that the Board of Patent Appeals overturn the final rejections of those claims at least for the reasons discussed below.

### **Claim Rejections - 35 U.S.C. § 103(a)**

Claims 1-4, 8, 9, 11, 21, and 23-25 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over *Karakashian, et al.* ("Karakashian," U.S. Pub. No. 2004/0064503) in view of *Kaler, et al.* ("Kaler," U.S. Pub. No. 2004/0199586) and further in view of *Kennedy* (U.S. Pat. No. 6,330,589). Applicant respectfully traverses.

As has been acknowledged by the Court of Appeals for the Federal Circuit, the U.S. Patent and Trademark Office ("USPTO") has the burden 35 U.S.C. § 103 to establish obviousness by showing objective teachings in the prior art or generally available knowledge of one of ordinary skill in the art that would lead that individual to the claimed invention. *In re Fine*, 837 F.2d 1071, 1074, 5 U.S.P.Q. 2d 1596, 1598 (Fed. Cir. 1988). The key to supporting an allegation of obviousness under 35 U.S.C. § 103 is the clear articulation of the reasons why the Examiner believes that claimed invention would have been obvious. See MPEP § 2141. As stated by the Supreme Court, "[r]ejections on obviousness cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." *KSR v. Teleflex*, 550 U.S. at \_\_\_, 82 USPQ2d at 1396 (quoting *In re Kahn*, 441 F.3d 977, 988, 78 USPQ2d 1329, 1336 (Fed. Cir. 2006)).

Applicant respectfully submits that the Examiner has not established with clearly articulated reasons that Applicant's claims are obvious in view of the prior art. Applicant discusses those claims in the following.

## **1. The Karakashian Disclosure**

Karakashian discloses a system and method for web service Java API-based invocation. *Karakashian*, Patent Application Title. As described by Karakashian, the system includes a web service container 108 that is served by a web container 100. *Karakashian*, paragraph 0032. The web container 100 includes a protocol adapter 102 that intercepts web service invocations from web services clients. *Karakashian*, paragraph 0032. As described by Karakashian in relation to Figure 2, the web service container 108 includes several components, including a container driver 200, interceptors 202-206, and an invocation handler 208. *Karakashian*, paragraph 0033.

During system operation, the protocol adapter 102 passes received message context (which contains a web service request) to the container driver 200 of the web service container 108. *Karakashian*, paragraph 0033. The container driver 200 sends the message context to the interceptors 202. *Karakashian*, paragraph 0033. After extracting the operation parameters from the message context, the container driver 200 submits an operation request to the invocation handler 208 for processing. *Karakashian*, paragraph 0033. The invocation handler 208 returns data to the container driver 200, which then sends a response to an outbound interceptor 202-206. *Karakashian*, paragraph 0033. Finally, the container driver 200 returns the response to the protocol adapter 102. *Karakashian*, paragraph 0033.

## **2. The Kaler Disclosure**

Kaler discloses a system and method for using expressive session information to represent communication sessions. *Kaler*, paragraph 0014.

## **3. The Kennedy Disclosure**

Kennedy discloses a system and method for managing conversation threads within an email context. *Kennedy*, column 1, lines 7-10; column 2, lines 47-49.

## **4. Applicant's Claims**

Independent claim 1 provides as follows:

1. A method for collecting data regarding a messaging session, the method comprising:

- intercepting an incoming message sent to a first network service;

- writing a session identifier to a thread-local variable, the session identifier identifying a messaging session to which the incoming message relates;

- storing in a database in relation to the session identifier session data relevant to the incoming message, the session data at least including a message received time;

- providing the incoming message to the first network service;

- sending an outgoing message from the network service to a second network service or a client;

- intercepting the outgoing message sent by the first network service;

- performing a thread-local variable lookup so as to retrieve the session identifier written to the thread-local variable;



storing in the database in relation to the session identifier session data relevant to the outgoing message, the session data at least including a message sent time; and

providing the outgoing message to the second network service or client.

In the final Office Action, the Examiner alleged that the cited references together disclose every limitation of claim 1. Applicant disagrees and discusses the Examiner's arguments in the following paragraphs.

As a first matter, the Examiner alleged that Karakashian discloses "writing a session identifier to a thread-local variable, the session identifier identifying a messaging session to which the incoming message relates". For support, the Examiner cited paragraph 0038 of the Karakashian reference. In that paragraph, Karakashian states:

An invocation context can be used, which is an inheritable thread-local object that can store arbitrary context data used in processing a web service request. The context can be available from various components of the architecture involved in the processing of the request and response. Typical data that might be stored in such a context are a conversation ID, a message sequence number, and a security token. A particular invocation handler can choose to make the invocation context available to the target component. This can allow application code to read and write to the invocation context.

*Karakashian*, paragraph 0038. Although Karakashian expresses in paragraph 0038 that "[a]n invocation context can be used" and further that the invocation context can include a "conversation ID," the fact remains that Karakashian does not disclose "writing" a

session identifier to a thread-local variable. If the Examiner disagrees, Applicant requests that the Examiner cite the particular lines of the Karakashian disclosure where such writing is described.

The Examiner further alleged that Karakashian discloses “performing a thread-local variable lookup so as to retrieve the session identifier written to the thread-local variable” prior to providing an outgoing message to a second network service or a client. For support, the Examiner identified paragraphs 0038 and 0107 of the Karakashian reference. Paragraph 0038 is reproduced above, and paragraph 0107 provides:

When the developer uses a static client application to invoke a web service, they use a strongly-typed Java interface, in contrast to a dynamic client where the developer indirectly references the web service operations and parameters. Using a dynamic client is analogous to looking up and invoking a method using the Java reflection APIs. The developer must include the web service-specific client JAR file in the CLASSPATH when statically invoking a web service. This JAR file includes the following classes and interfaces:

*Karakashian*, paragraph 0107. As can be appreciated from paragraphs 0038 and 0107, Karakashian does not in fact disclose “performing a thread-local variable lookup so as to retrieve the session identifier written to the thread-local variable”. Again, although Karakashian mentions invocation context and a conversation ID, nothing within paragraphs 0038 and 0107 (or the remainder of the Karakashian disclosure) discloses or suggests such an action. If the Examiner disagrees, Applicant requests that the Examiner cite the particular lines of the Karakashian disclosure where a thread-local variable lookup is described as being performed.

Later in the final Office Action, the Examiner admitted that Karakashian does not disclose or suggest “storing in a database in relation to the session identifier session data relevant to the incoming message, the session data at least including a message received time”. To account for that deficiency of the Karakashian reference, the Examiner cited the Kaler reference, which was alleged to disclose such storing. In response, Applicant notes that Kaler both fails to disclose storing session data in a database, as well as session data that includes “a message received time”. Applicant has carefully reviewed each of paragraphs 0006, 0016, 0017, 0019, and 0041 of the Kaler reference, which were cited by the Examiner, and found no reference to storing session information or that such storage information comprises a message received time. Therefore, it appears clear that Kaler does not in fact disclose those aspects of claim 1. If the Examiner disagrees, Applicant requests that the Examiner cite the particular lines of the Kaler disclosure where such aspects are actually described.

Still later in the final Office Action, the Examiner admitted that Karakashian does not disclose or suggest “storing in the database in relation to the session identifier session data relevant to the outgoing message, the session data at least including a message sent time”. To account for that deficiency of the Karakashian reference, the Examiner cited the Kennedy reference, which was alleged to disclose such storing. In response, Applicant notes that the Kennedy reference fails to disclose storing “a message sent time”, i.e., the time a message was sent from a “first network service” to a “second network service or client”. Although Kennedy mentions a “posted or sent time,” Kennedy indicates that the time pertains to a time at which a message (i.e., email message) is “posted to a database” See *Kennedy*, column 3, lines 35-38. Therefore,

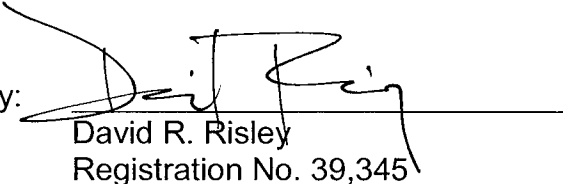
Kennedy's "posted or sent time" is not the same as the "message sent time" recited by Applicant. That Kennedy chose the same terminology to refer to the time a message is posted to a database does not change that fact.

In view of the above, it is clear that claim 1 and its dependents are allowable over the applied references. Applicant therefore requests that the rejections be withdrawn. Given that independent claim 21 contains limitations similar to those discussed above, Applicant further submits that claim 21 and its dependents are likewise allowable and that the rejections against those claims should likewise be withdrawn.

### **VIII. Conclusion**

In summary, it is Applicant's position that Applicant's claims are patentable over the applied prior art references and that the rejection of these claims should be withdrawn. Appellant therefore respectfully requests that the Board of Appeals overturn the Examiner's rejection and allow Applicant's pending claims.

Respectfully submitted,

By:   
David R. Risley  
Registration No. 39,345

**Claims Appendix under 37 C.F.R. § 41.37(c)(1)(viii)**

The following are the claims that are involved in this Appeal.

1. A method for collecting data regarding a messaging session, the method comprising:

intercepting an incoming message sent to a first network service;

writing a session identifier to a thread-local variable, the session identifier identifying a messaging session to which the incoming message relates;

storing in a database in relation to the session identifier session data relevant to the incoming message, the session data at least including a message received time;

providing the incoming message to the first network service;

sending an outgoing message from the network service to a second network service or a client;

intercepting the outgoing message sent by the first network service;

performing a thread-local variable lookup so as to retrieve the session identifier written to the thread-local variable;

storing in the database in relation to the session identifier session data relevant to the outgoing message, the session data at least including a message sent time; and

providing the outgoing message to the second network service or client.

2. The method of claim 1, wherein intercepting an incoming message comprises intercepting an extensible markup language (XML) message wrapped in a simple object access protocol (SOAP) envelope.

3. The method of claim 1, wherein intercepting an incoming message comprises intercepting a service request.

4. The method of claim 1, wherein intercepting an incoming message comprises intercepting a service response.

5-7. (Canceled)

8. The method of claim 1, wherein writing a session identifier to a thread-local variable comprises writing the session identifier to a thread-local variable using a simple object access protocol (SOAP) message handler.

9. The method of claim 1, wherein storing session data regarding the incoming message further comprises storing a source name of the sender of the message, a message type, and a destination name of the intended recipient.

10. (Canceled)

11. The method of claim 1, wherein storing session data regarding the outgoing message further comprises storing the source name of the sender of the message, the message type, and the destination name of the intended recipient.

12-20. (Canceled)

21. A physical computer-readable medium that stores a system for collecting data regarding a messaging session, the system comprising:

logic configured to intercept an incoming message directed at a network service;

logic configured to identify a session identifier that identifies a messaging session to which the incoming message relates;

logic configured to write the session identifier to a thread-local variable;

logic configured to store in a database in relation to the session identifier session data relevant to the incoming message, the session data at least including a message received time;

logic configured to intercept an outgoing message sent by the network service;

logic configured to perform a thread-local variable lookup so as to retrieve the session identifier written to the thread-local variable; and

logic configured to store in the database in relation to the session identifier session data relevant to the outgoing message, the session data at least including a message received time.

22. (Canceled)

23. The computer-readable medium of claim 21, wherein the logic configured to write the session identifier to a thread-local variable comprises a message handler.



24. The computer-readable medium of claim 23, wherein the message handler comprises a simple object access protocol (SOAP) message handler.

25. The computer-readable medium of claim 21, wherein the logic configured to store session data regarding the incoming message is further configured to store a source name of the sender of the message, a message type, and a destination name of the intended recipient, and wherein the logic configured to store session data regarding the outgoing message is further configured to store the source name of the sender of the message, the message type, and the destination name of the intended recipient.

26-36. (Canceled)

**Evidence Appendix under 37 C.F.R. § 41.37(c)(1)(ix)**

There is no extrinsic evidence to be considered in this Appeal. Therefore, no evidence is presented in this Appendix.

**Related Proceedings Appendix under 37 C.F.R. § 41.37(c)(1)(x)**

There are no related proceedings to be considered in this Appeal. Therefore, no such proceedings are identified in this Appendix.